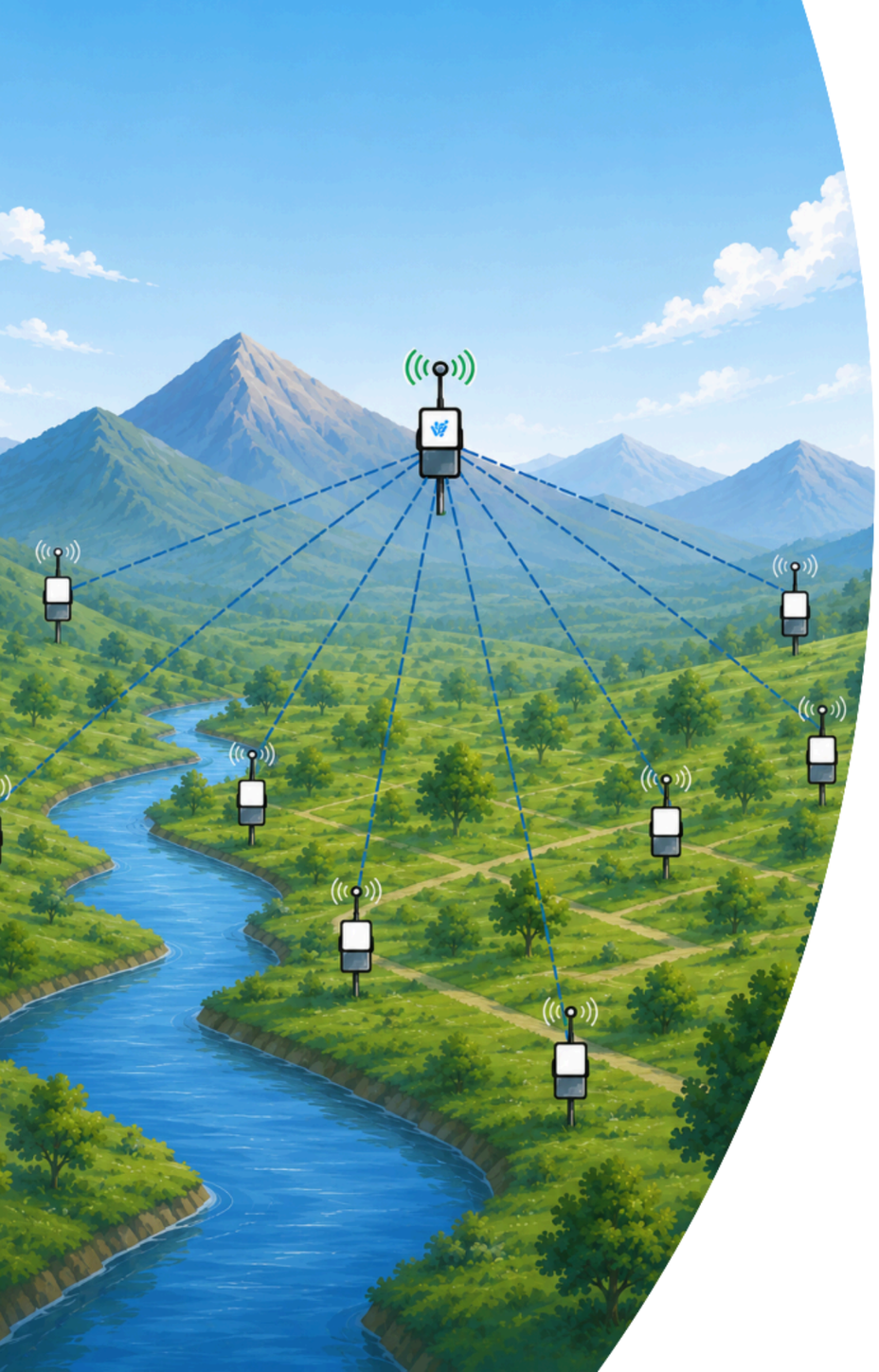


B.SC. GRADUATION PROJECT — ELECTRICAL ENGINEERING

Application on Ad Hoc Networks Using Wireless Sensor Networks

Al-Azhar University · Faculty of Engineering

Supervised by: **Prof. Abdellatif I. Moustafa** · July 2026

Indonesia, 2015

When a Forest Fire Became a Global Disaster

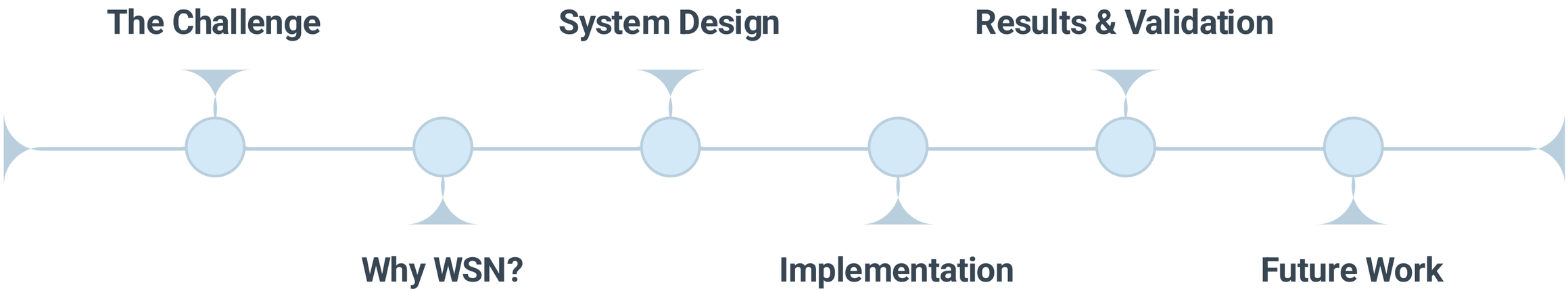
In 2015, Indonesia faced one of the worst environmental catastrophes in modern history exposing a critical gap in early fire detection and real-time environmental monitoring.

2.6M	$16B	1.6B	500K+
Hectares Burned	Economic Loss	Tons of CO_2 Emitted	Respiratory Illness Cases

Core Question: Can wireless sensor networks detect forest fires before they become uncontrollable?

From Problem to Solution

Our project follows a structured engineering methodology — from identifying the monitoring gap to validating a real-world WSN-based solution.



The Challenge

Why Traditional Monitoring Falls Short

🌲 Environmental

- Forest fires
- Smart agriculture
- Air & water quality

🏥 Healthcare

- Patient vital signs
- Elderly monitoring
- Remote healthcare

🏭 Industrial

- Patient vital signs
- Elderly monitoring
- Remote healthcare

🌆 Smart Cities

- Bridges & tunnels
- Pipelines
- Traffic & public safety

Conclusion: These environments require **reliable, continuous, and remote monitoring** — a capability only wireless sensor networks can deliver at scale.

Why Wireless Sensor Networks?

A Wireless Sensor Network transforms raw environmental data into actionable intelligence — autonomously and in real time.

Remote Deployment

Works without existing infrastructure

Low Power Consumption

Designed for long-term operation

Self-Organizing Network

Automatic network formation

Scalability

Easy to expand coverage

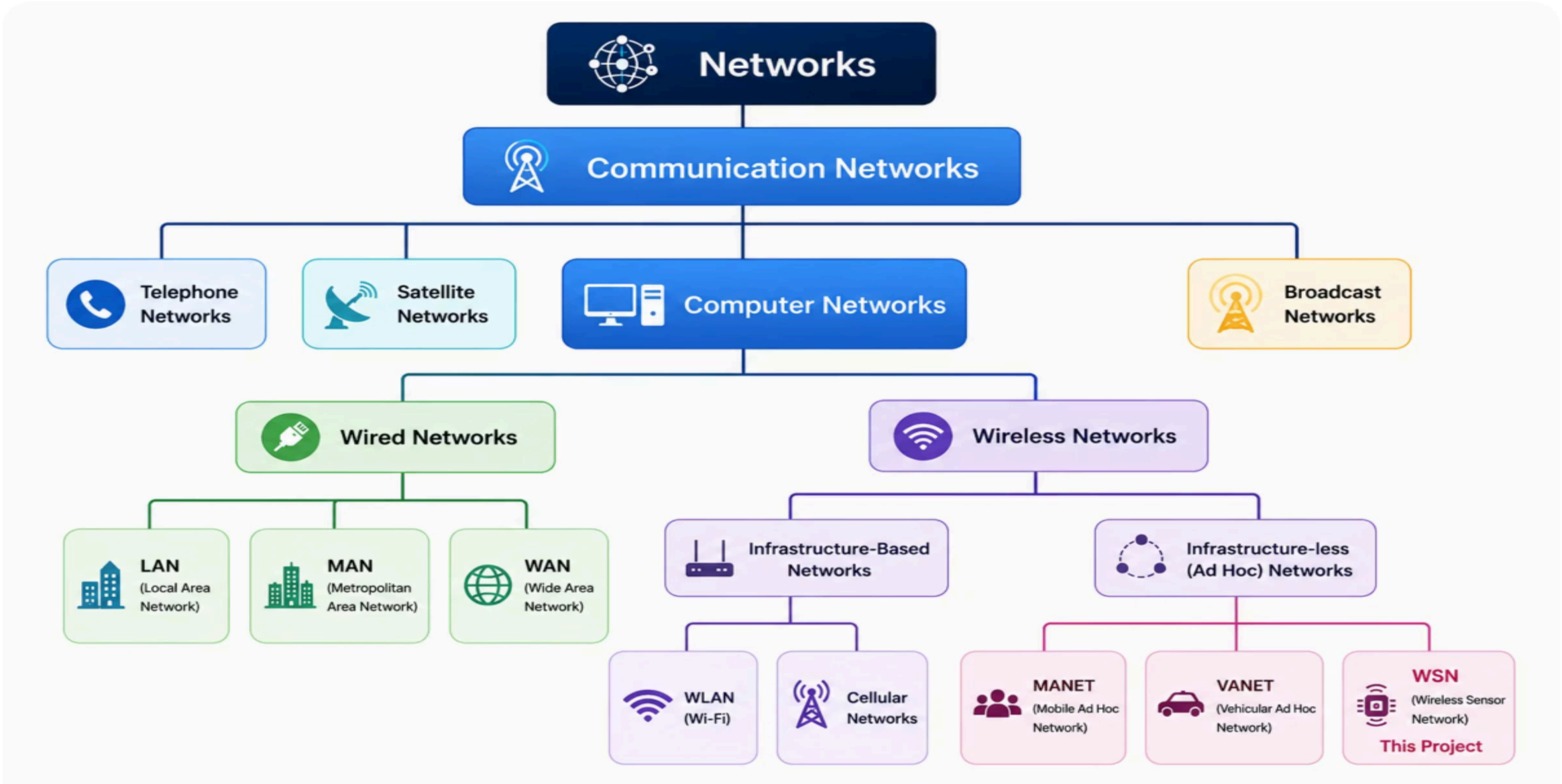
Real-Time Monitoring

Continuous environmental sensing

Reliable Sensing

Multi-sensor data collection

From Network to WSN



Wireless Sensor Networks in Ad Hoc Environments

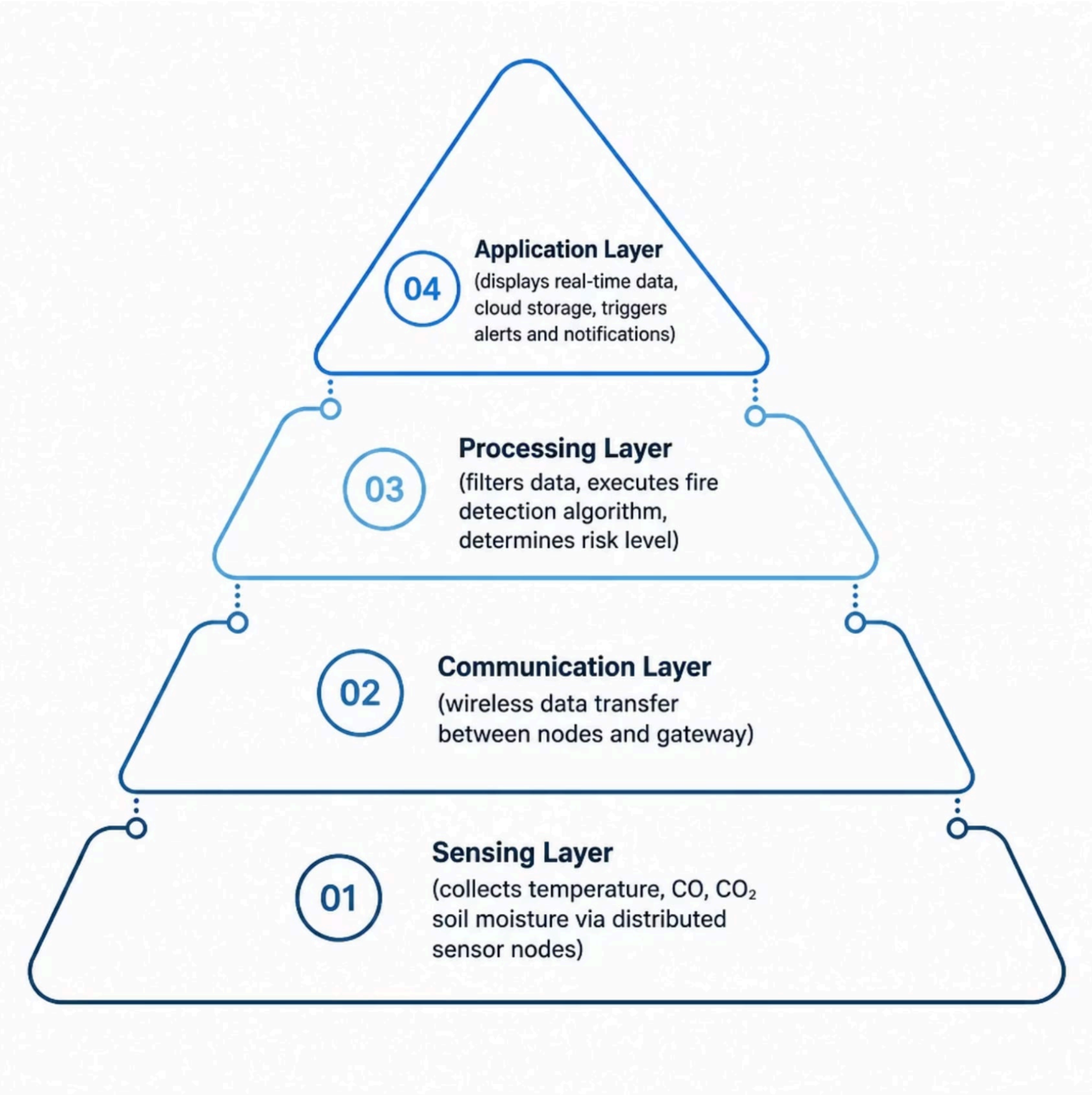
WSNs are a specialized form of Ad Hoc Networks, optimized for sensing and monitoring applications in dynamic and infrastructure-less environments.

WSN (Wireless Sensor Networks)	VS	Ad Hoc Networks
Low Power Consumption Designed for energy efficiency to maximize network lifetime.		**Higher Power Usage** General-purpose communication with higher energy requirements.
Sensor-Based Architecture Built to collect, process, and transmit data from physical environment.		**General Communication** Focused on data exchange between devices, not specifically on sensing.
Self-Organization Nodes automatically organize and adapt to changes in the network.		**Dynamic Topology** Supports mobility and rapid topology changes for general communication needs.
Long Network Lifetime Optimized for long-term monitoring missions with minimal maintenance.		**Shorter Lifetime** Not optimized for long-term, continuous monitoring applications.
Real-Time Monitoring Designed for continuous sensing and real-time environmental data delivery.		**Purpose** Used in applications like military, emergency communication, and general networking.

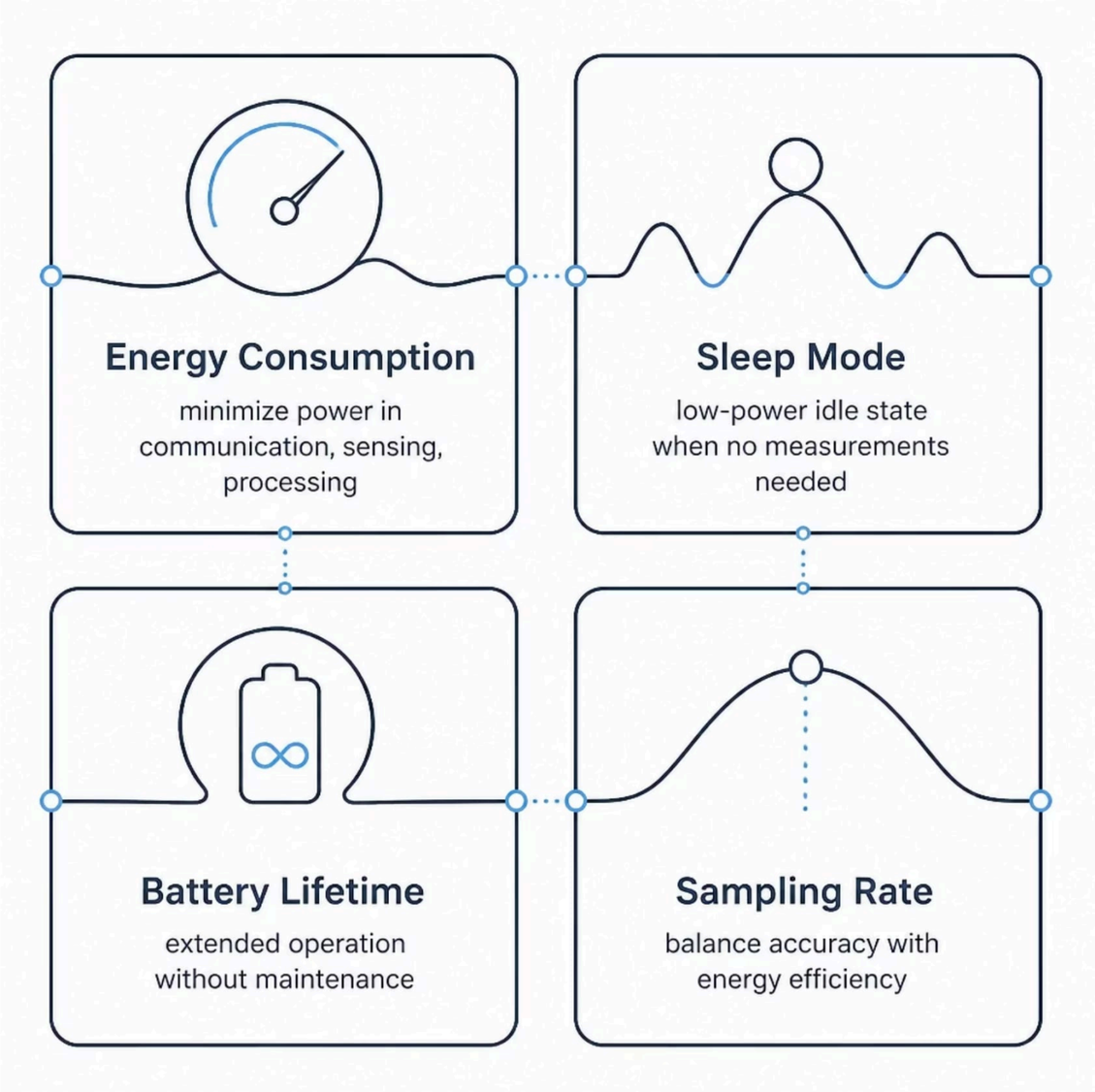
For our **forest fire monitoring** application, WSN provides the **specialized capabilities** required for reliable sensing and data collection in ad hoc environments.

WSN Architecture

The WSN architecture consists of four functional layers that work together to collect, transmit, process, and utilize environmental data for real-time monitoring and decision-making.



Power Management in WSN



Sensor nodes are battery-powered and deployed in remote locations where battery replacement is difficult. Optimizing energy consumption ensures reliable, long-term operation.

Efficient power management extends network lifetime, reduces maintenance costs, and ensures continuous monitoring for early fire detection in inaccessible forest environments.

WSN Hardware Platform

Why LabVIEW?

LabVIEW is the core software platform, providing an integrated environment for data acquisition, processing, visualization, storage, and communication — seamlessly integrated with NI hardware.

Acquire

Receives real-time temperature, CO, CO_2 , and soil moisture readings from the NI WSN Gateway.

Process

Validates readings and executes the fire detection algorithm.

Visualize

Displays live sensor values and fire risk through an interactive GUI.

Store

Records measurements locally and in the cloud for historical analysis.

Communicate

Exchanges data with Python backend via TCP and syncs with Firebase.

LabVIEW Development Environment

LabVIEW provides a graphical programming environment that simplifies the development of data acquisition and monitoring systems.

Front Panel

User Interface — Displays real-time sensor readings, graphs, controls, and indicators for live monitoring.

Block Diagram

Program Logic — Graphical source code implementing data acquisition, processing, and control algorithms.

Data Acquisition (DAQ)

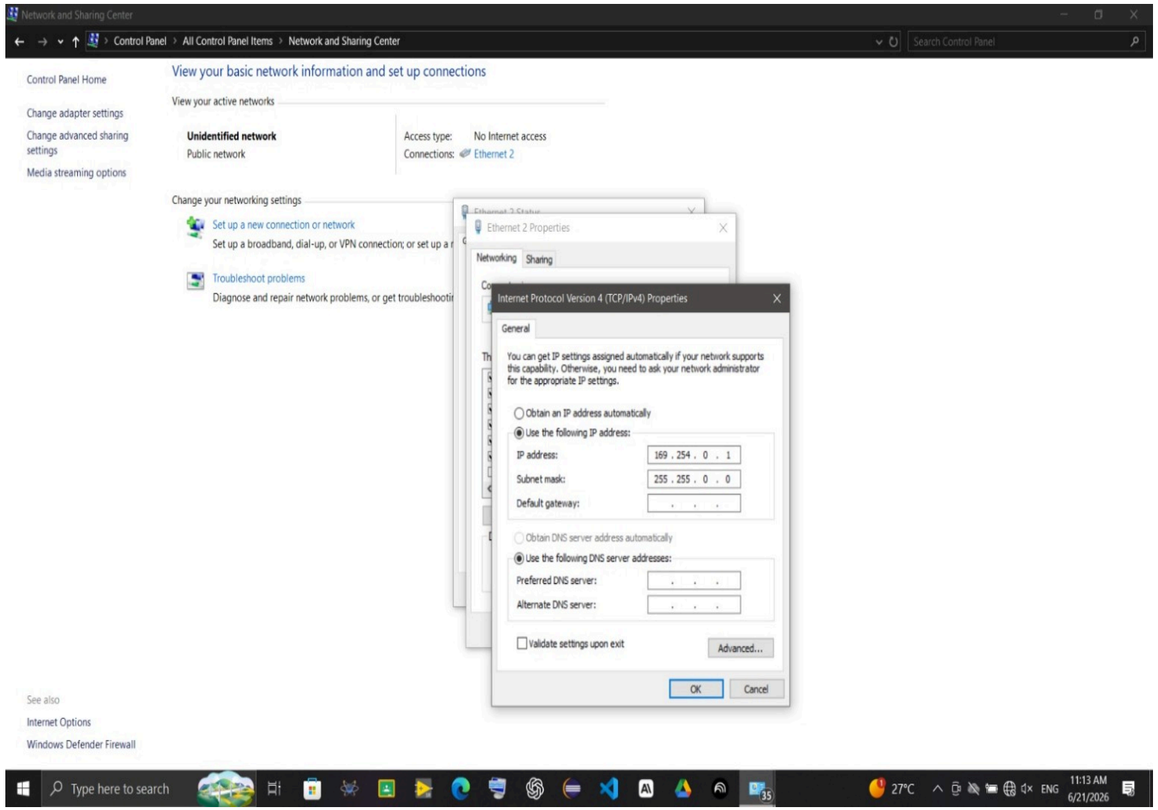
Sensor Data Collection — Interfaces with NI hardware to acquire real-time measurements from wireless sensor nodes.

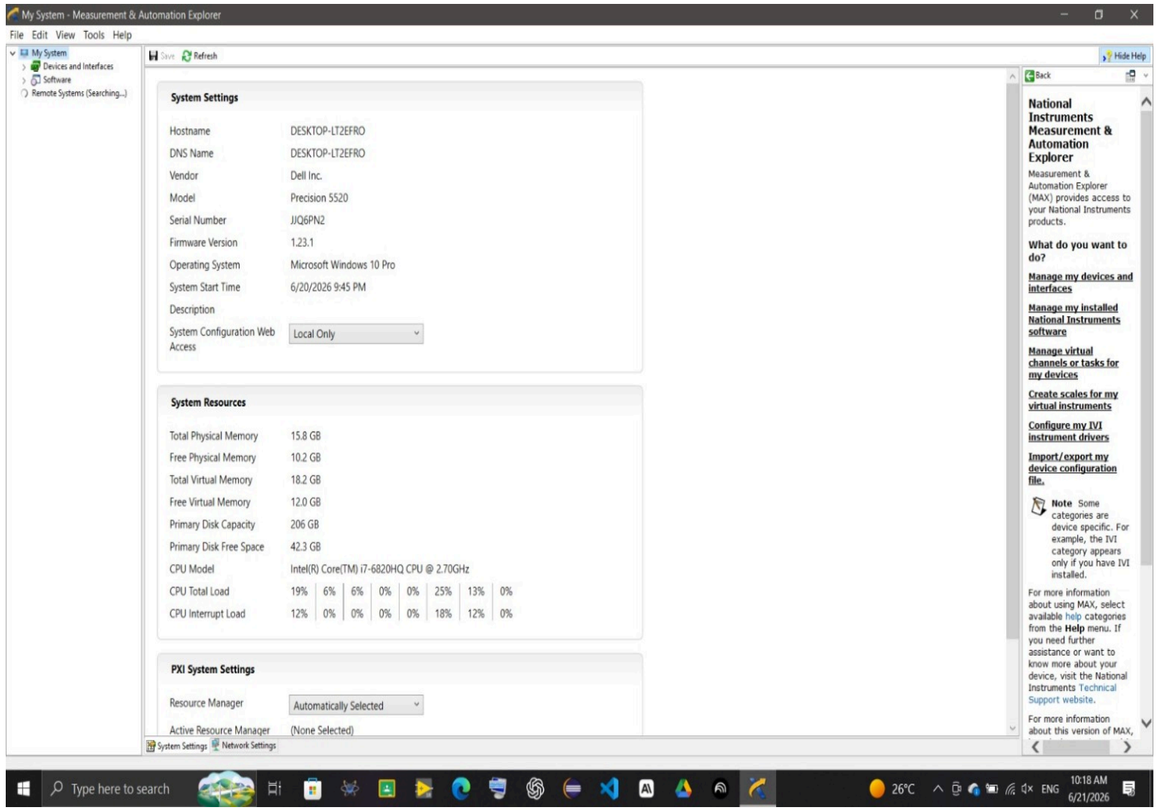
VI & SubVI

Modular Building Blocks — Virtual Instruments combine Front Panel and Block Diagram; SubVIs enable code reuse and maintainability.

Gateway Configuration

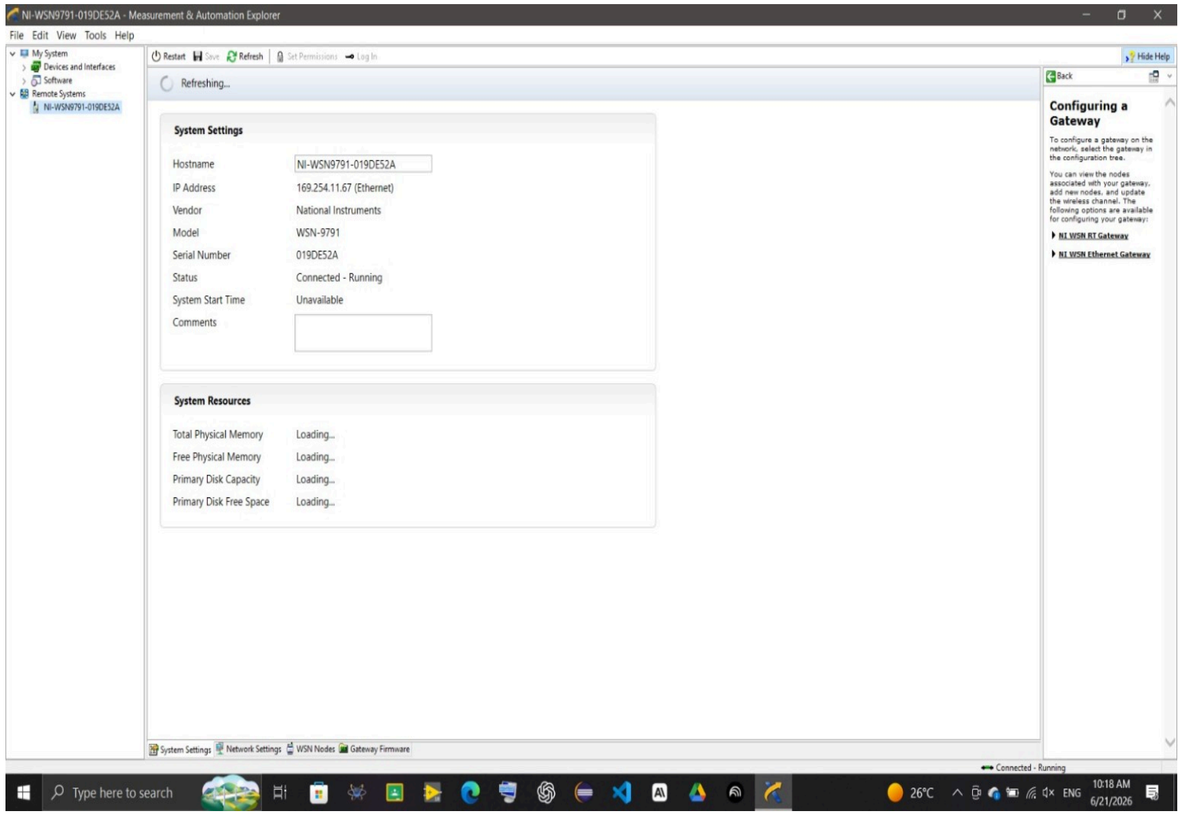
- Configure Ethernet connection.
- Configure IPv4 in the same subnet.
- Launch NI MAX.
- Verify the host system.
- Detect the gateway (Connected – Running).

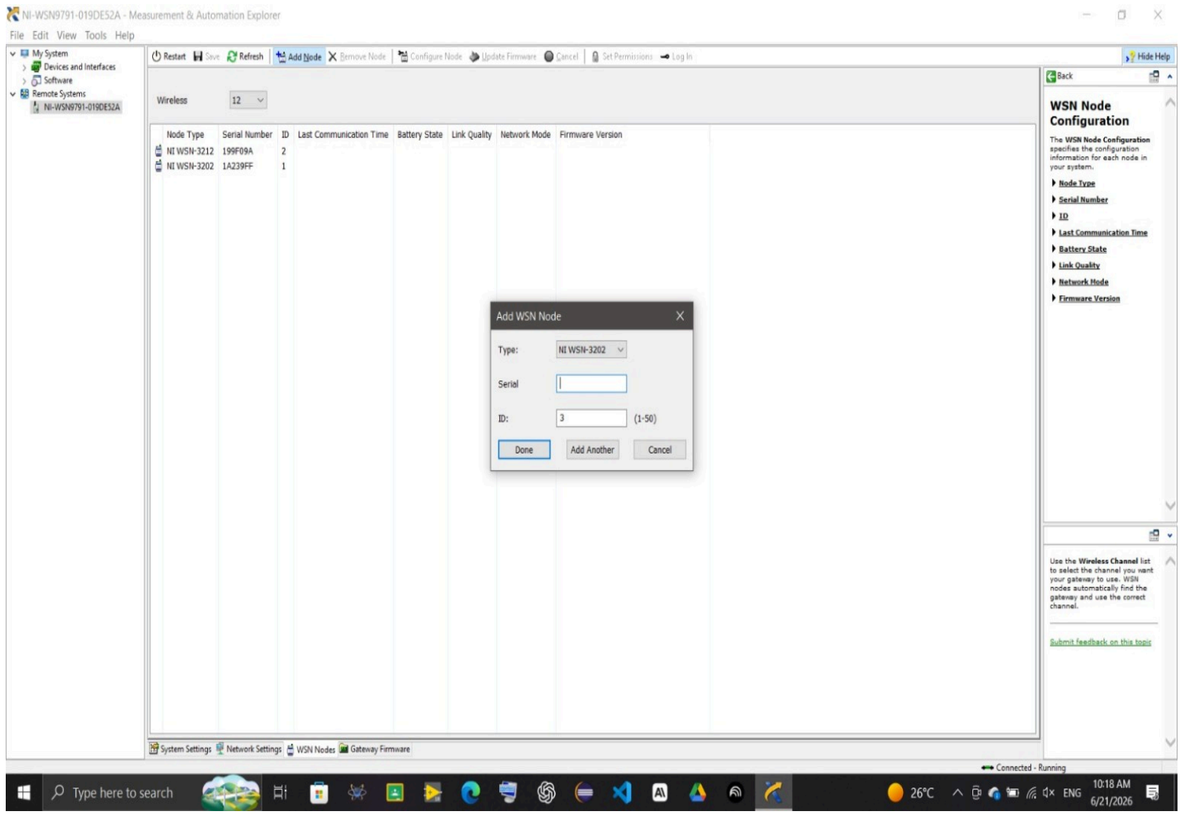

Network and Sharing Center
Control Panel > All Control Panel Items > Network and Sharing Center
Search Control Panel
Control Panel Home
Change adapter settings
Change advanced sharing settings
Media streaming options
View your basic network information and set up connections
View your active networks
Unidentified network
Public network
Access type: No Internet access
Connections: Ethernet 2
Change your networking settings
Set up a new connection or network
Set up a broadband, dial-up, or VPN connection; or set up a
Troubleshoot problems
Diagnose and repair network problems, or get troubleshooti
See also
Internet Options
Windows Defender Firewall
Ethernet 2 Properties
Networking
Sharing
Internet Protocol Version 4 (TCP/IPv4) Properties
General
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.
Obtain an IP address automatically
Use the following IP address:
IP address: 169 . 254 . 0 . 1
Subnet mask: 255 . 255 . 0 . 0
Default gateway:
Obtain DNS server address automatically
Use the following DNS server addresses:
Preferred DNS server:
Alternate DNS server:
Validate settings upon exit
Advanced...
OK
Cancel
Type here to search
27°C
ENG
11:13 AM
6/21/2026


My System - Measurement & Automation Explorer
File Edit View Tools Help
My System
Devices and Interfaces
Software
Remote Systems (Searching...)
Save
Refresh
Hide Help
Back
System Settings
Hostname DESKTOP-LT2EFRO
DNS Name DESKTOP-LT2EFRO
Vendor Dell Inc.
Model Precision 5520
Serial Number JJQ6PN2
Firmware Version 1.23.1
Operating System Microsoft Windows 10 Pro
System Start Time 6/20/2026 9:45 PM
Description
System Configuration Web Access Local Only
System Resources
Total Physical Memory 15.8 GB
Free Physical Memory 10.2 GB
Total Virtual Memory 18.2 GB
Free Virtual Memory 12.0 GB
Primary Disk Capacity 206 GB
Primary Disk Free Space 42.3 GB
CPU Model Intel(R) Core(TM) i7-6820HQ CPU @ 2.70GHz
CPU Total Load 19% 6% 6% 0% 0% 25% 13% 0%
CPU Interrupt Load 12% 0% 0% 0% 0% 18% 12% 0%
PXI System Settings
Resource Manager Automatically Selected
Active Resource Manager (None Selected)
System Settings
Network Settings
National Instruments Measurement & Automation Explorer
Measurement & Automation Explorer (MAX) provides access to your National Instruments products.
What do you want to do?
Manage my devices and interfaces
Manage my installed National Instruments software
Manage virtual channels or tasks for my devices
Create scales for my virtual instruments
Configure my IVI instrument drivers
Import/export my device configuration file.
Note Some categories are device specific. For example, the IVI category appears only if you have IVI installed.
For more information about using MAX, select available help categories from the Help menu. If you need further assistance or want to know more about your device, visit the National Instruments Technical Support website.
For more information about this version of MAX,
Type here to search
26°C
ENG
10:18 AM
6/21/2026

Wireless Sensor Nodes & LabVIEW Integration

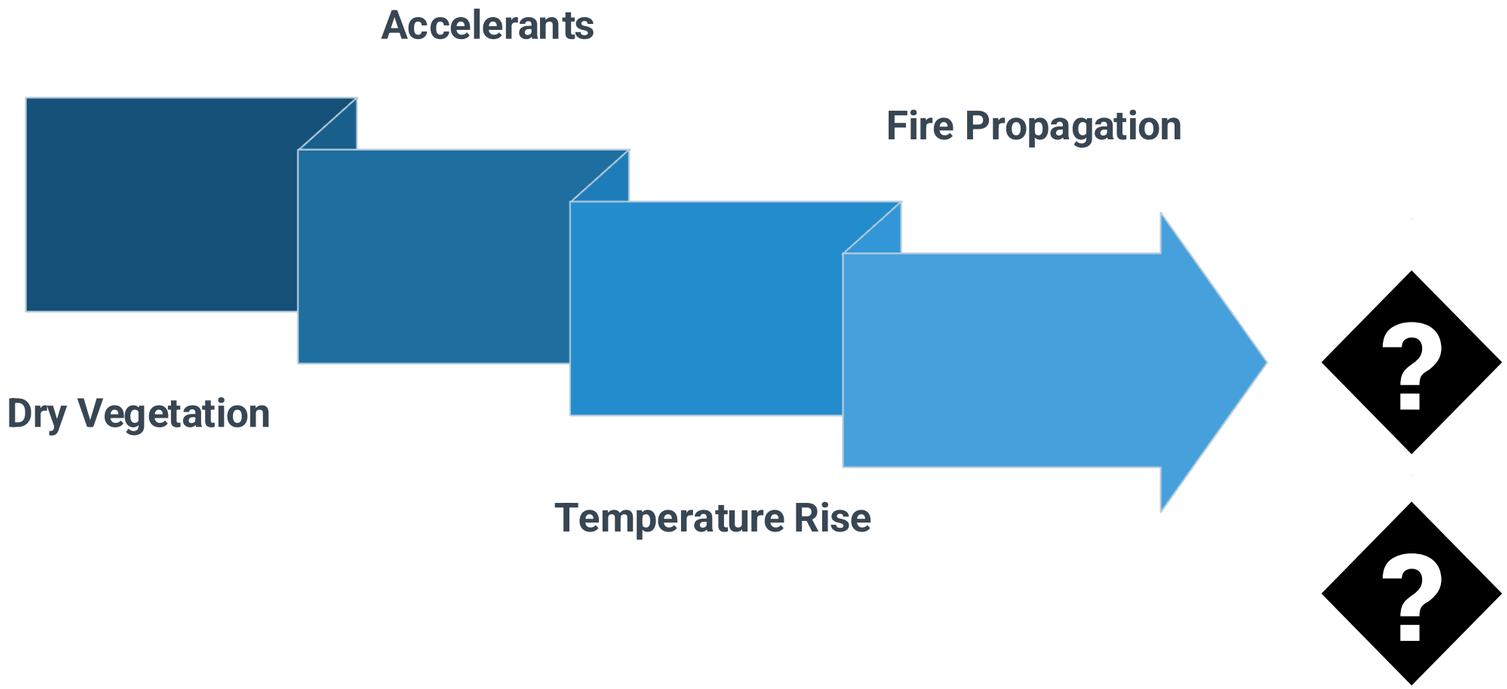
- Open the WSN Nodes tab.
- Detect and configure the sensor nodes.
- Verify successful connection.
- Ready for LabVIEW integration.





How Does a Forest Fire Develop?

Forest fires develop through a sequence of environmental changes that can be detected before the fire spreads enabling early-stage identification of fire risks.



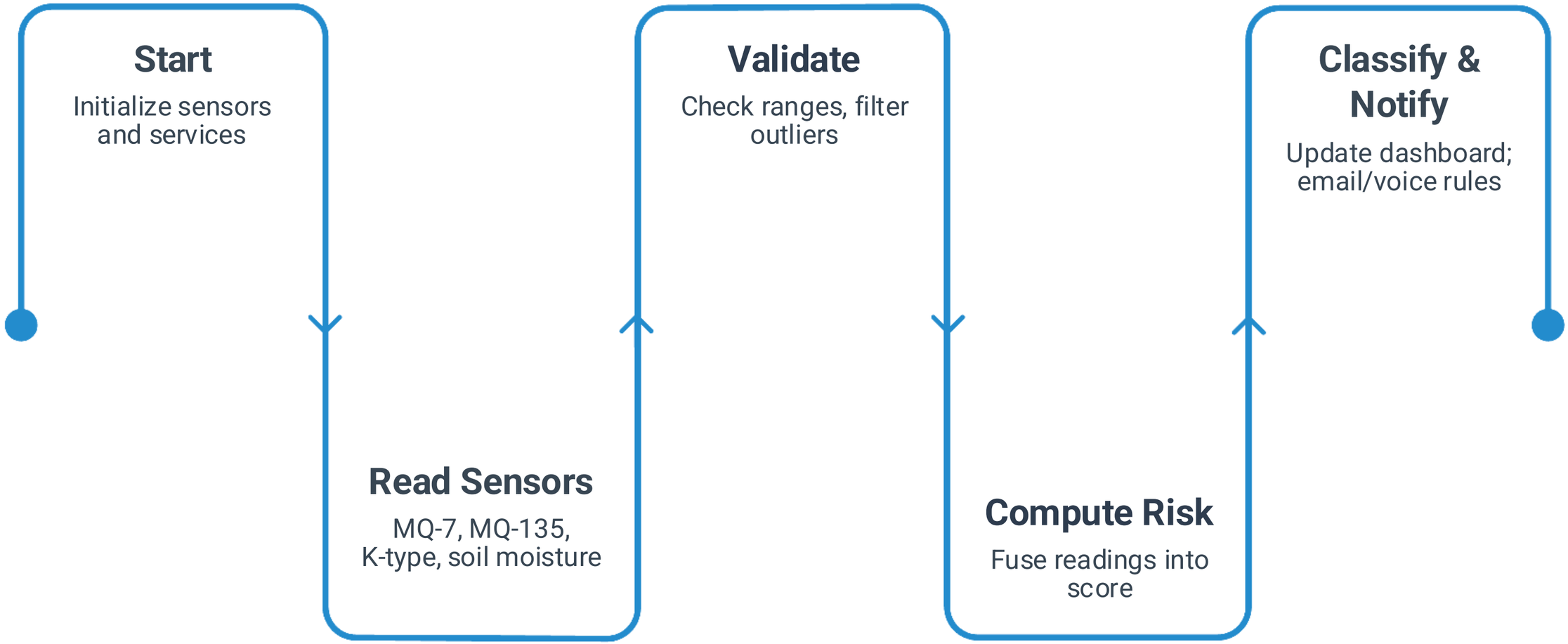
Key Insight: By monitoring soil moisture, temperature, CO, and CO_2 , the proposed system can detect abnormal conditions before a small ignition develops into a large-scale forest fire.

Sensors Used in the Proposed System

Sensor	Measured Parameter	Measurement Range	Purpose
MQ-7	Carbon Monoxide (CO)	0– 500 ppm (scaled)	Detect combustion gases and CO spikes
MQ-135	Air Quality & Smoke	0– 10,000 ppm	Detect smoke, polluted air, hazardous VOCs
Thermocouple K-type	Temperature	0– 482 ° C	Monitor ambient and hotspot temperatures
Soil Moisture Sensor	Soil Moisture	0– 100% (after calibration)	Monitor ground moisture to refine fire-risk models

Multi-sensor data fusion improves fire detection accuracy and reduces false alarms by combining gas, particulate/air-quality, temperature, and ground-moisture indicators.

Fire Detection Algorithm



The algorithm continuously reads all sensors, validates values, computes a composite fire-risk score using calibrated thresholds and fusion logic, classifies into discrete risk bands, updates cloud/dashboard, and triggers notifications based on severity and connectivity status.

Fire Risk Classification

Level 0 — SAFE

Normal environmental conditions. No action required.

Level 1 — LOW WARNING

Minor abnormalities detected. Increased observation; email alert sent.

Level 2 — MEDIUM DANGER

Multiple fire indicators present. Email and voice call alerts activated.

Level 3 — HIGH DANGER

Strong evidence of an active or imminent fire. Immediate escalation.

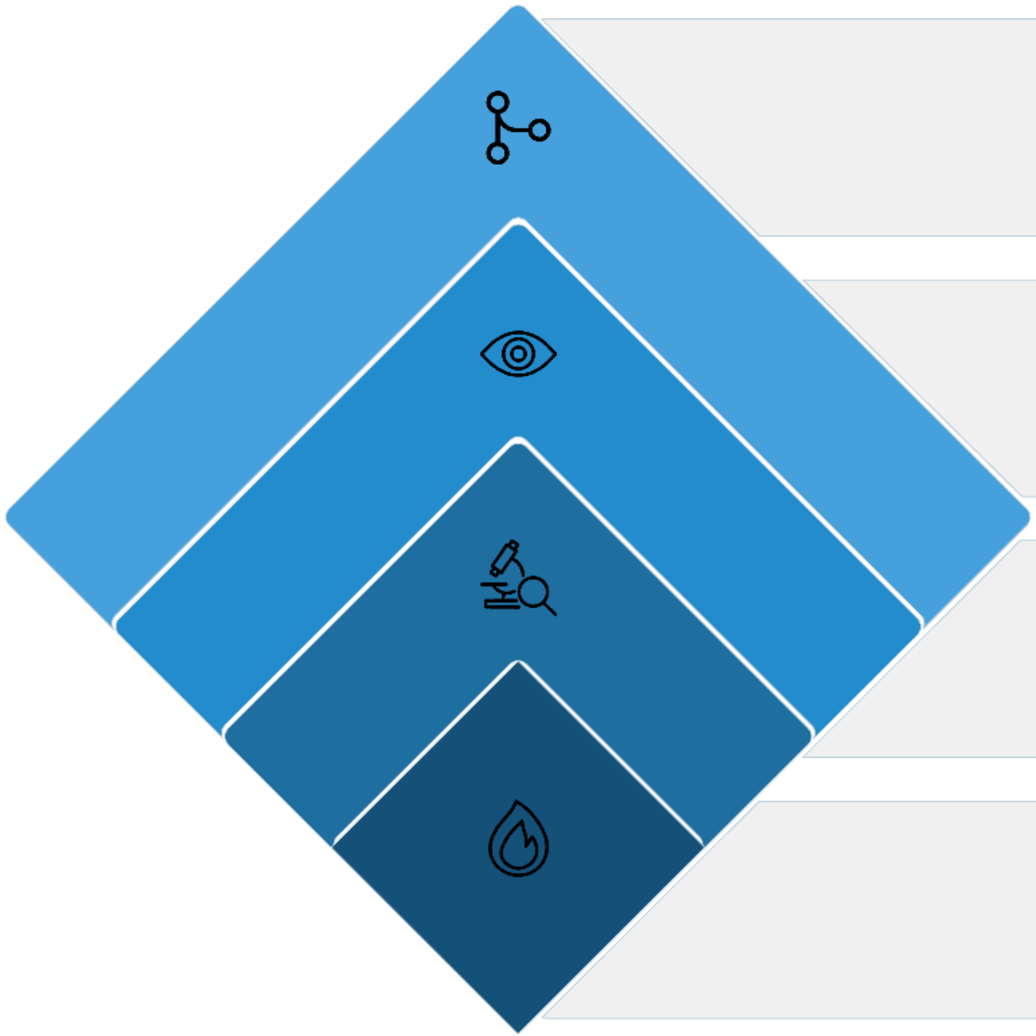
Level 4 — CRITICAL

Active fire detected or highly probable — emergency response required.

Level	Status	Firebase	Email	Voice Call
0	Safe	Yes	No	No
1	Low Warning	Yes	Yes	No
2	Medium Danger	Yes	Yes	Yes
3	High Danger	Yes	Yes	Yes
4	Critical	Yes	Yes	Yes

The fire risk level is determined by combining temperature, carbon monoxide, air quality, and soil moisture measurements.

Automatic Notification System



Levels 1– 4

Log to Firebase and dashboard

Level 0

Continue monitoring

Evaluate Risk

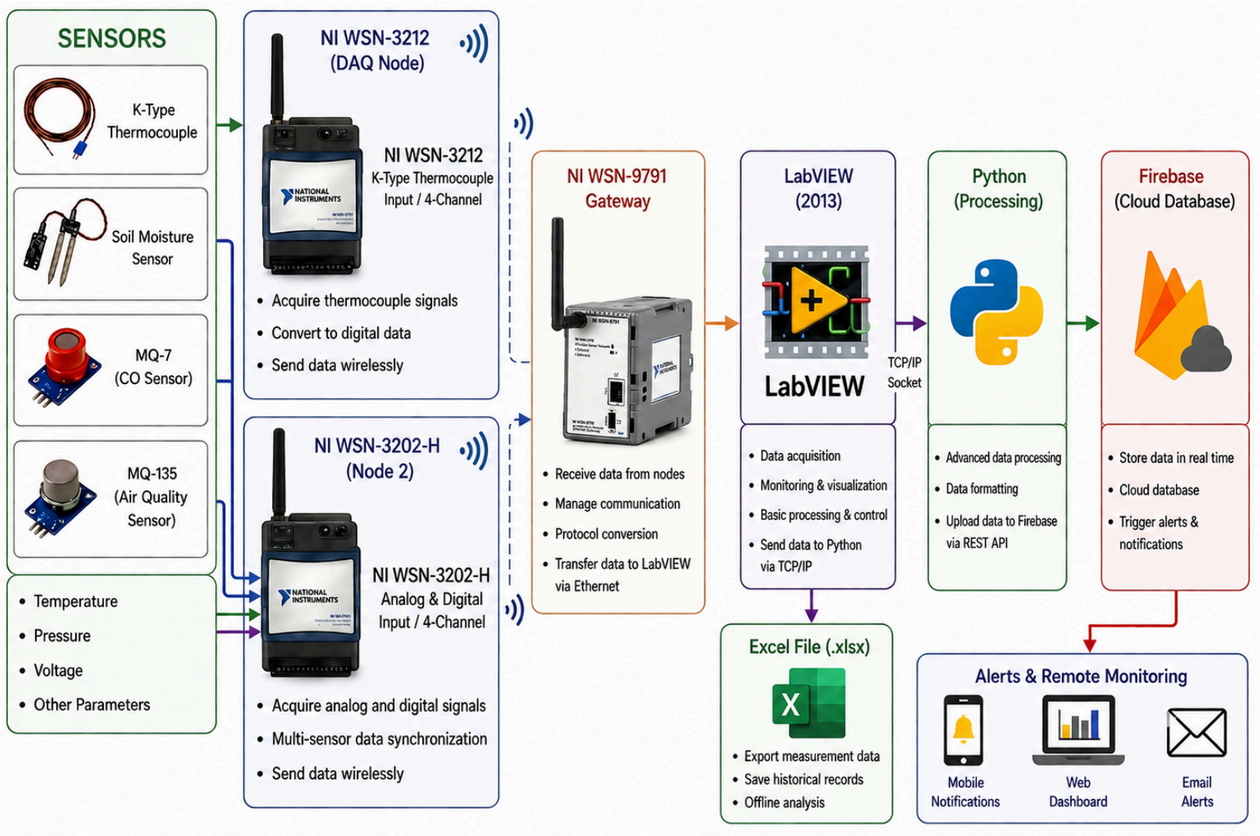
Determine level 0– 4

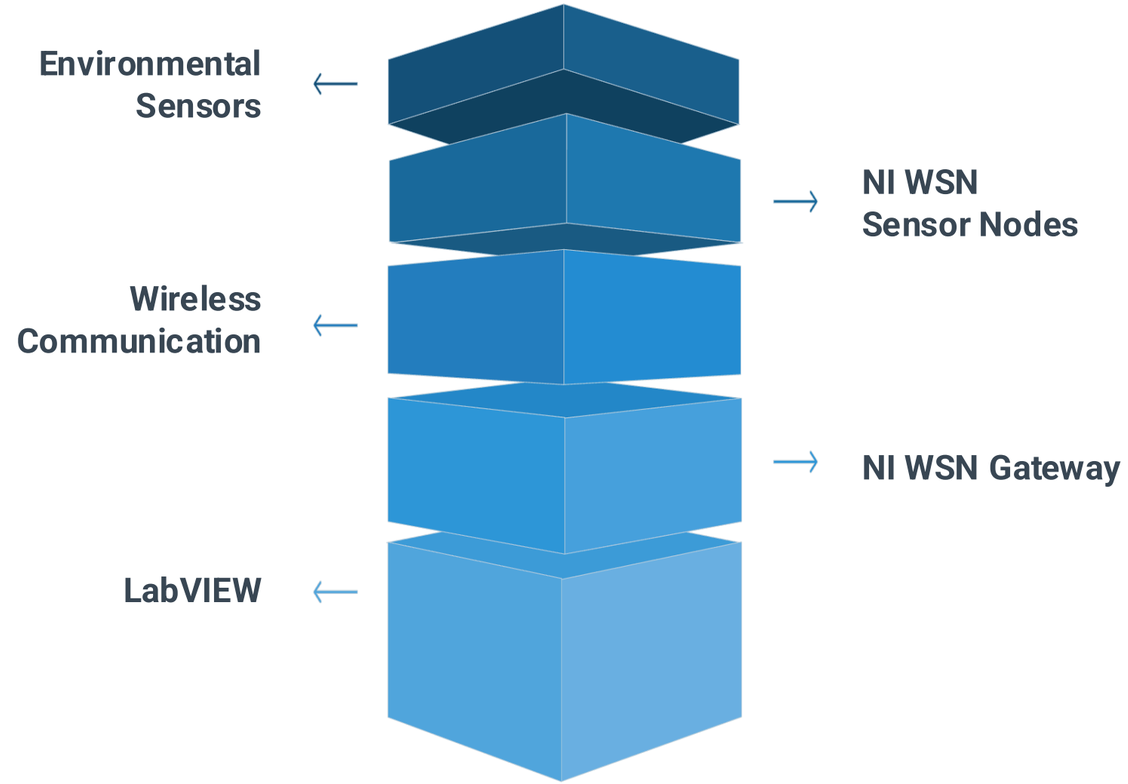
Fire Detected

Automatic alert triggered

The notification logic routes messages based on evaluated risk: low warnings generate email notifications only; medium and above escalate to both email and voice calls. All events are logged to Firebase and reflected on the dashboard for operators.

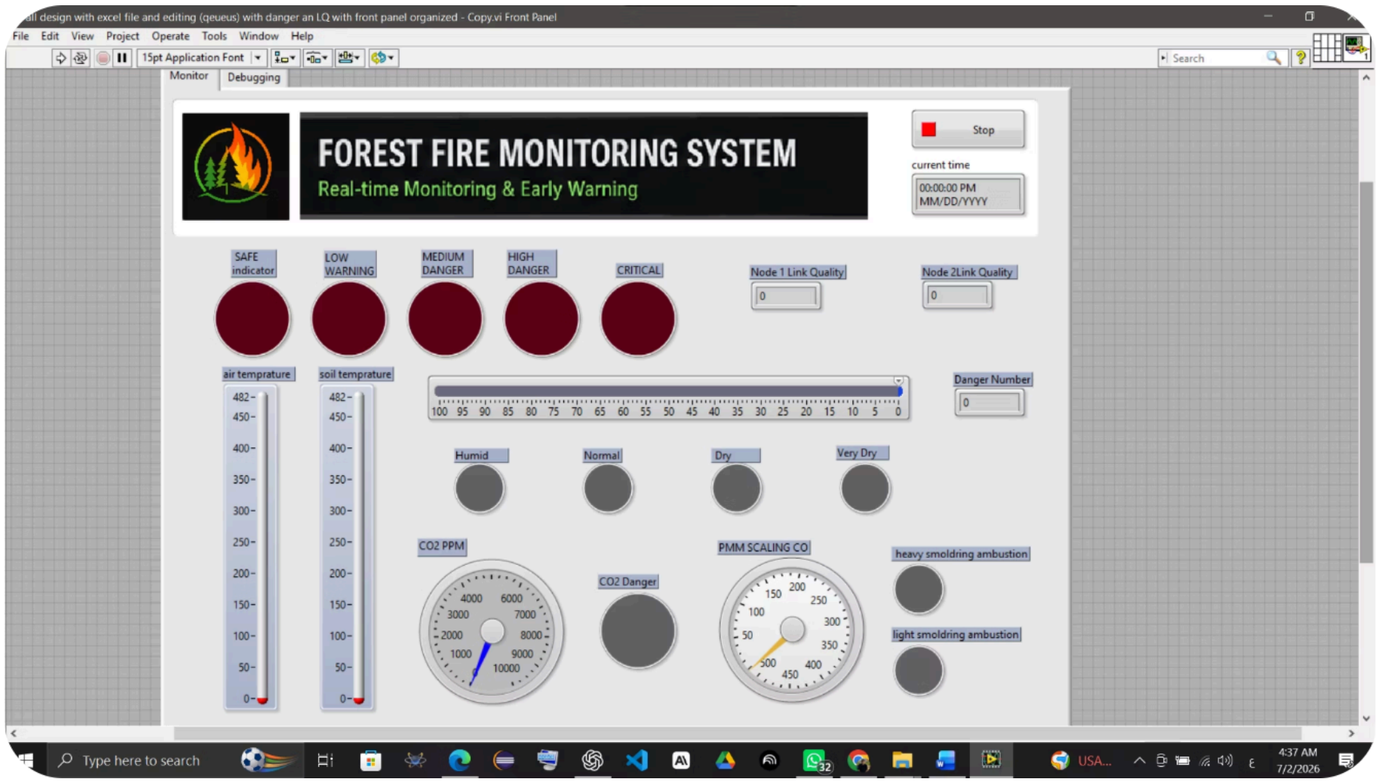
System Implementation



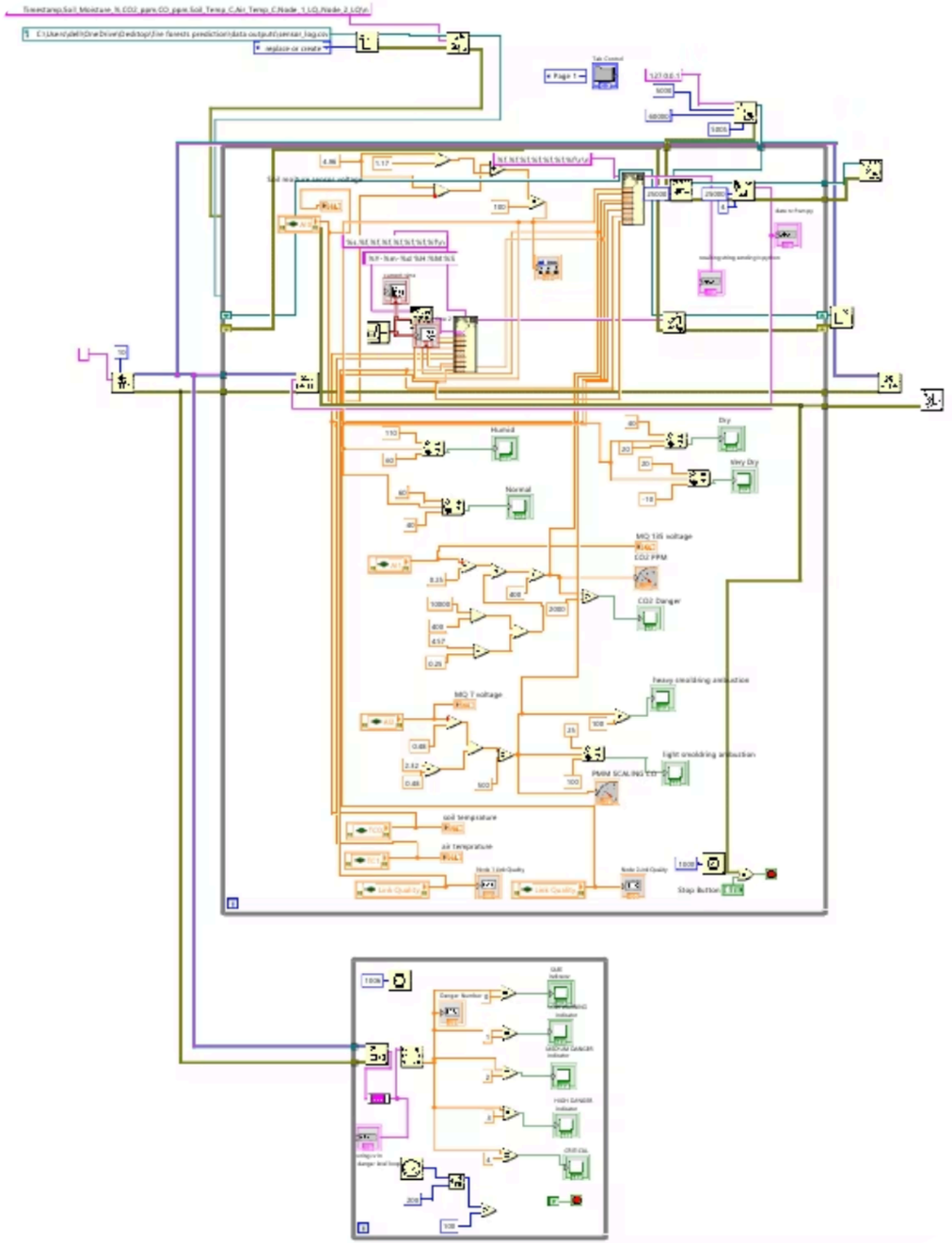


The NI WSN-3202 acquires MQ-7, MQ-135, and soil moisture sensors.
The NI WSN-3212 connects to two K-type thermocouples.
Both nodes transmit wirelessly to the NI WSN Gateway,
Which forwards data to LabVIEW for real-time acquisition and processing.

LabVIEW Implementation



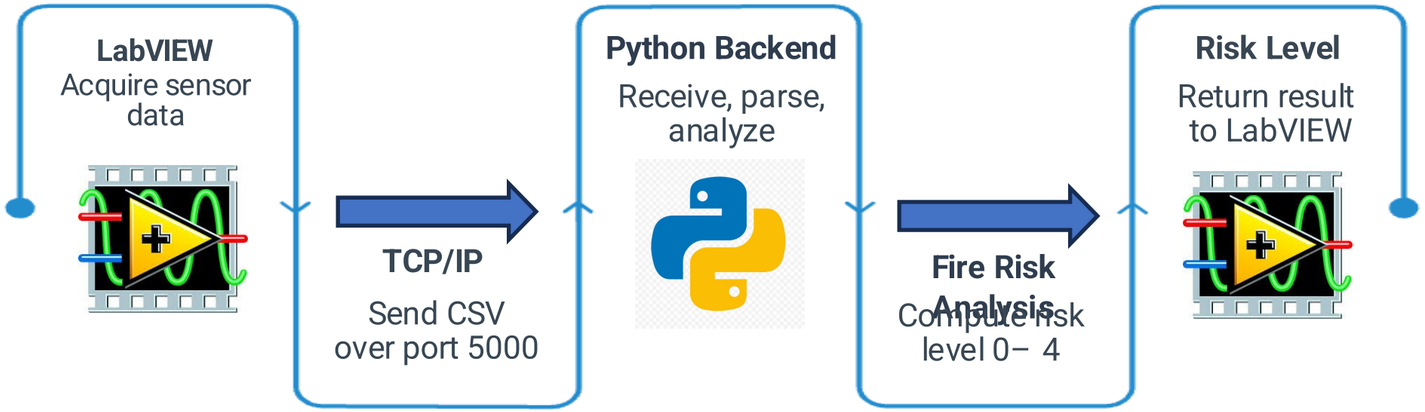
Block Diagram System Logic and Data Processing

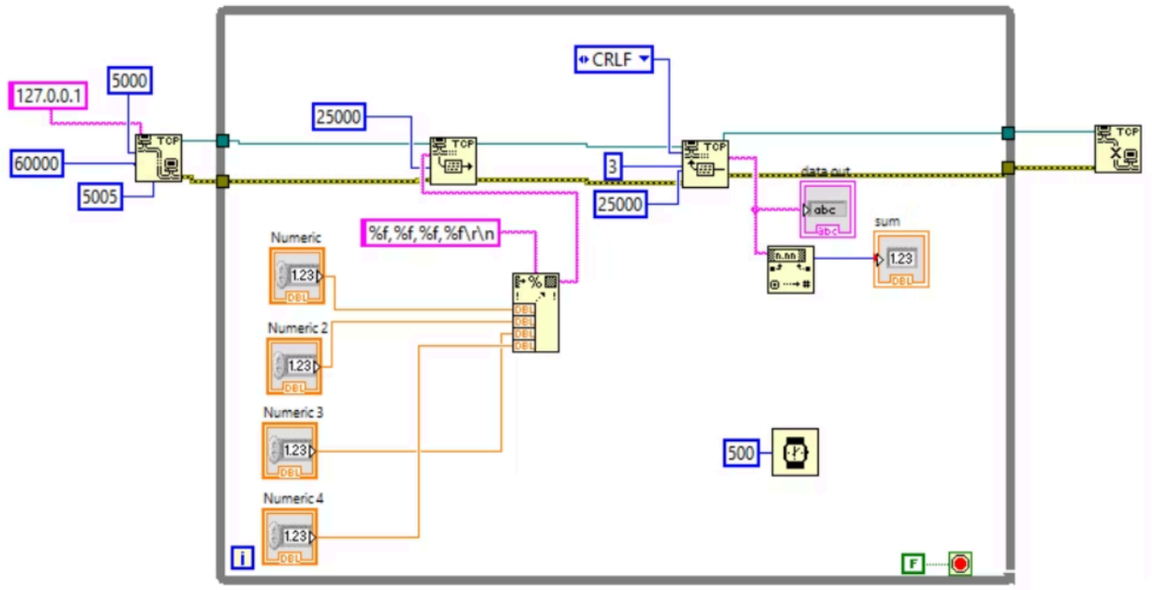


Front Panel Real-Time Monitoring Interface

LabVIEW serves as the **central** control platform for data acquisition, visualization, **TCP/IP** communication with the Python backend, and data logging.

TCP/IP Communication





LabVIEW TCP Communication Block Diagram

Communication Details

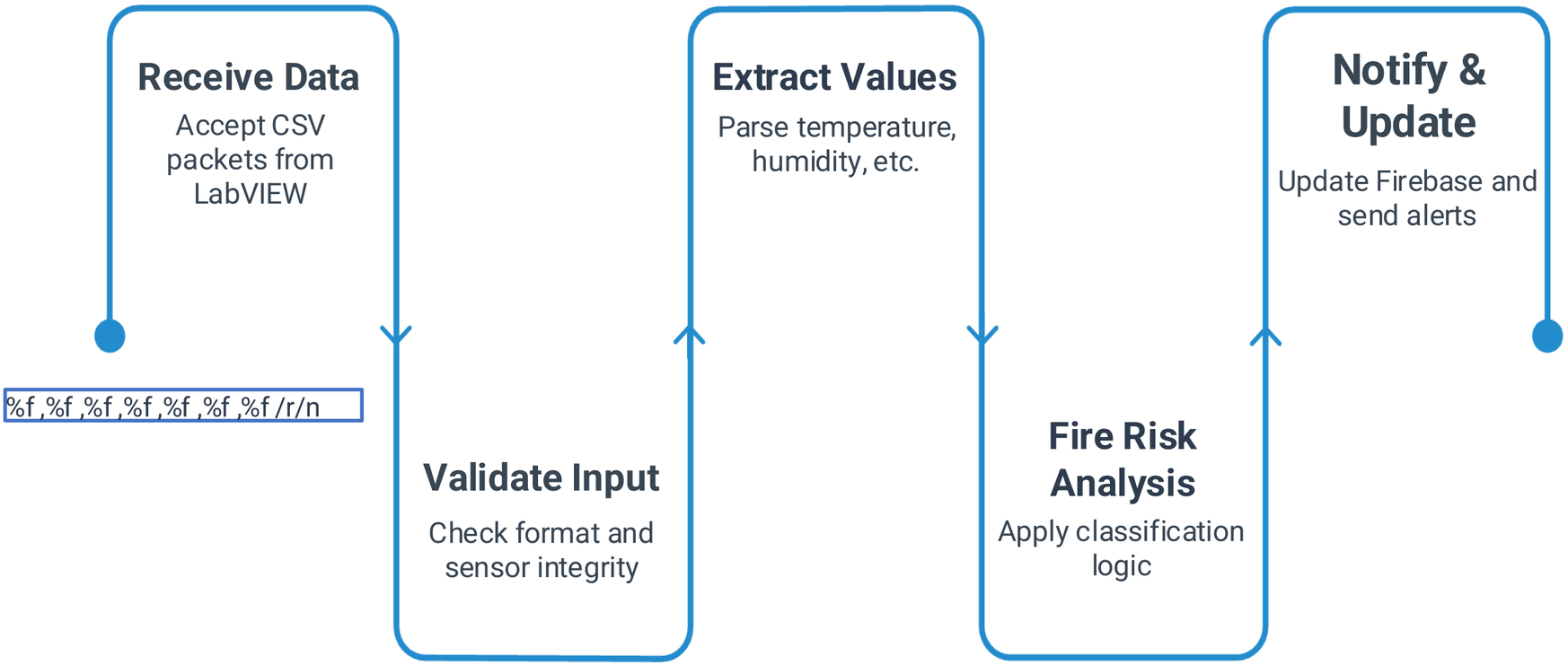
Protocol	TCP/IP
LabVIEW → Python	Port 5000
Python → LabVIEW	Port 5000
Data Format	CSV
Message Terminator	CRLF (\r\n)
Communication Mode	Full Duplex

Real-time TCP/IP communication enables reliable bidirectional data exchange between LabVIEW and the Python backend for fire risk assessment.

Example Data Packet

23.6,145,410,58\r\n ↓ Risk Level = 2

Python Backend Processing



Python Responsibilities

- **TCP Server Communication**
- **Data Validation**
- **Fire Risk Classification**
- **Firebase Integration**
- **Email Notification**
- **Voice Call Alert**
- **Return Risk Level to LabVIEW**

Python acts as the intelligent backend: it receives validated sensor CSV packets, applies the risk model, updates cloud services, triggers alerts according to policy, and returns the discrete risk level to LabVIEW for display and logging.

System Outputs & Data Integration

Email Notification

Immediate alerts

Python Backend

Computes Risk Level 0– 4

Alert System

Notification orchestration

Risk Level Distribution

LabVIEW Dashboard

Real-time Monitoring

Firebase Realtime Database

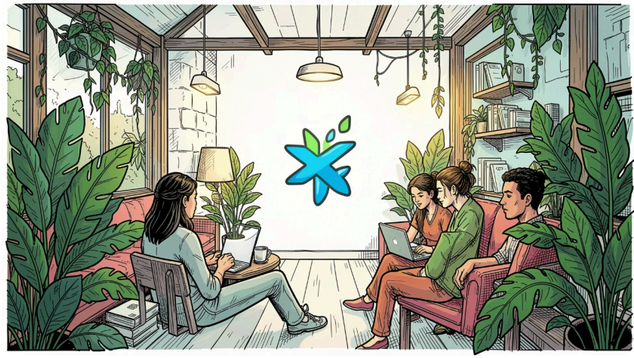
Cloud Storage

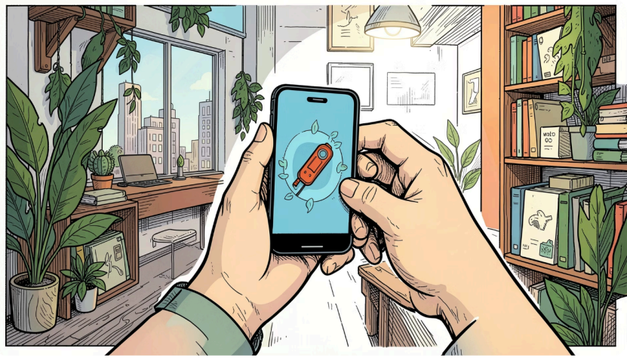
Excel Logging

Historical records

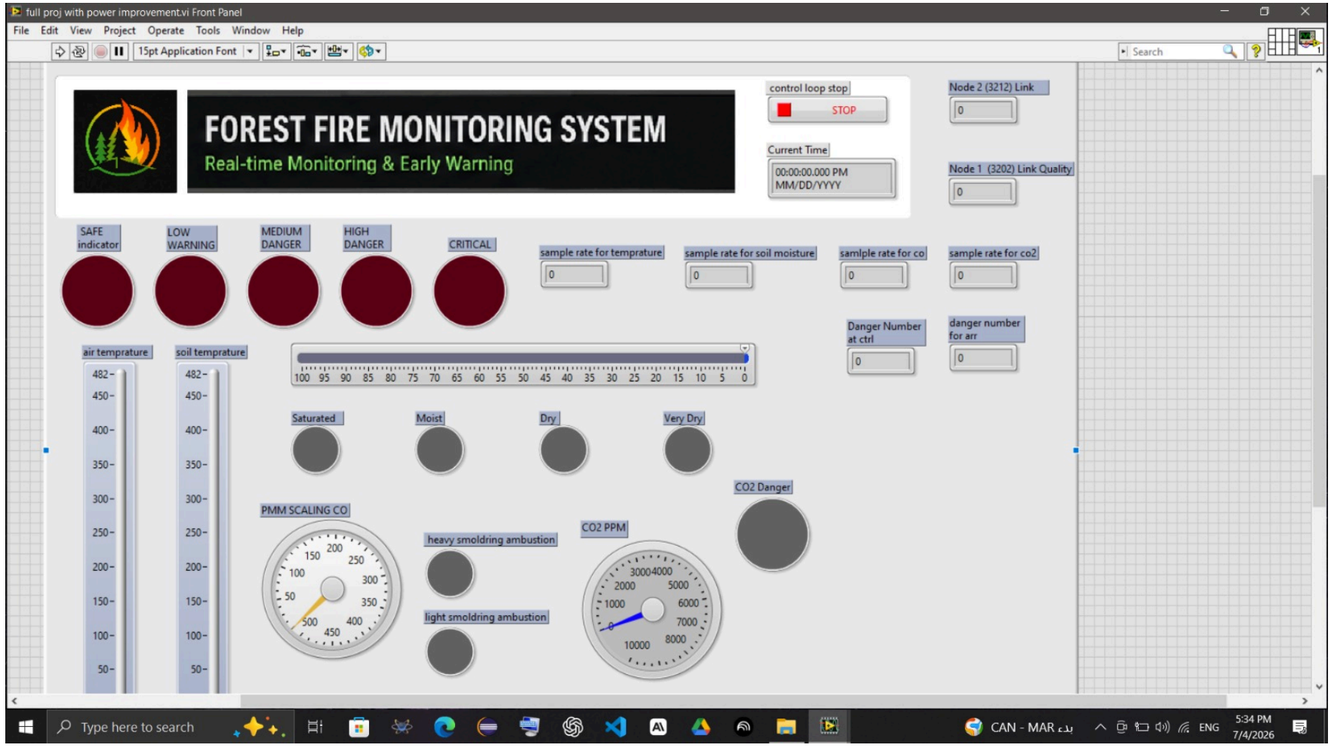
System Outputs

- Risk Level Display
- Dashboard Visualization
- Firebase Update
- Email Notification
- Voice Call Alert
- Excel Data Logging





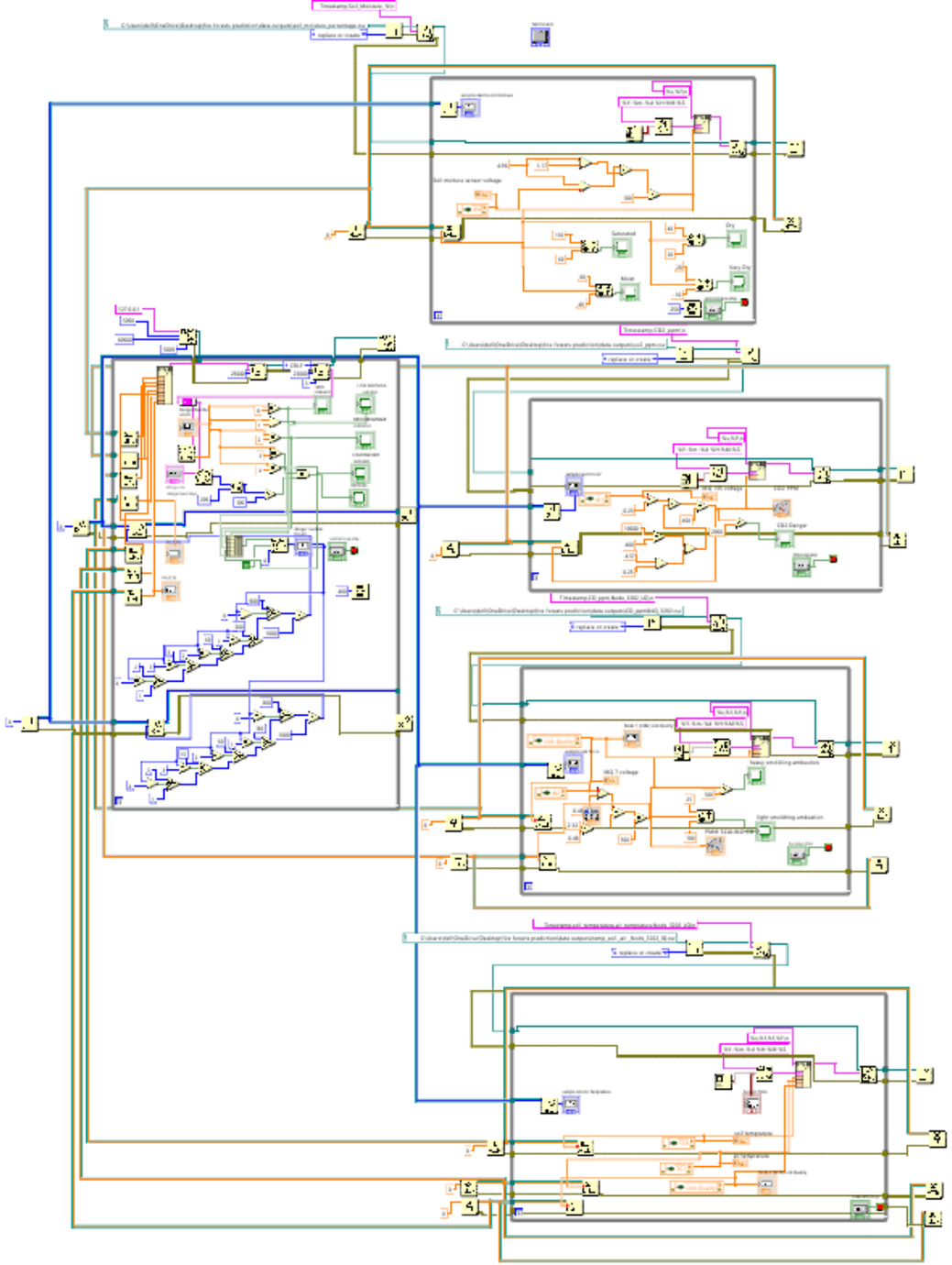
Enhanced System Architecture



Block Diagram System Logic and Data Processing

Architecture Advantages:

Five independent parallel loops
Fully decoupled architecture
Queue-based FIFO communication
Notifier-based adaptive control
Fault isolation between loops
Easy sensor expansion
High reliability

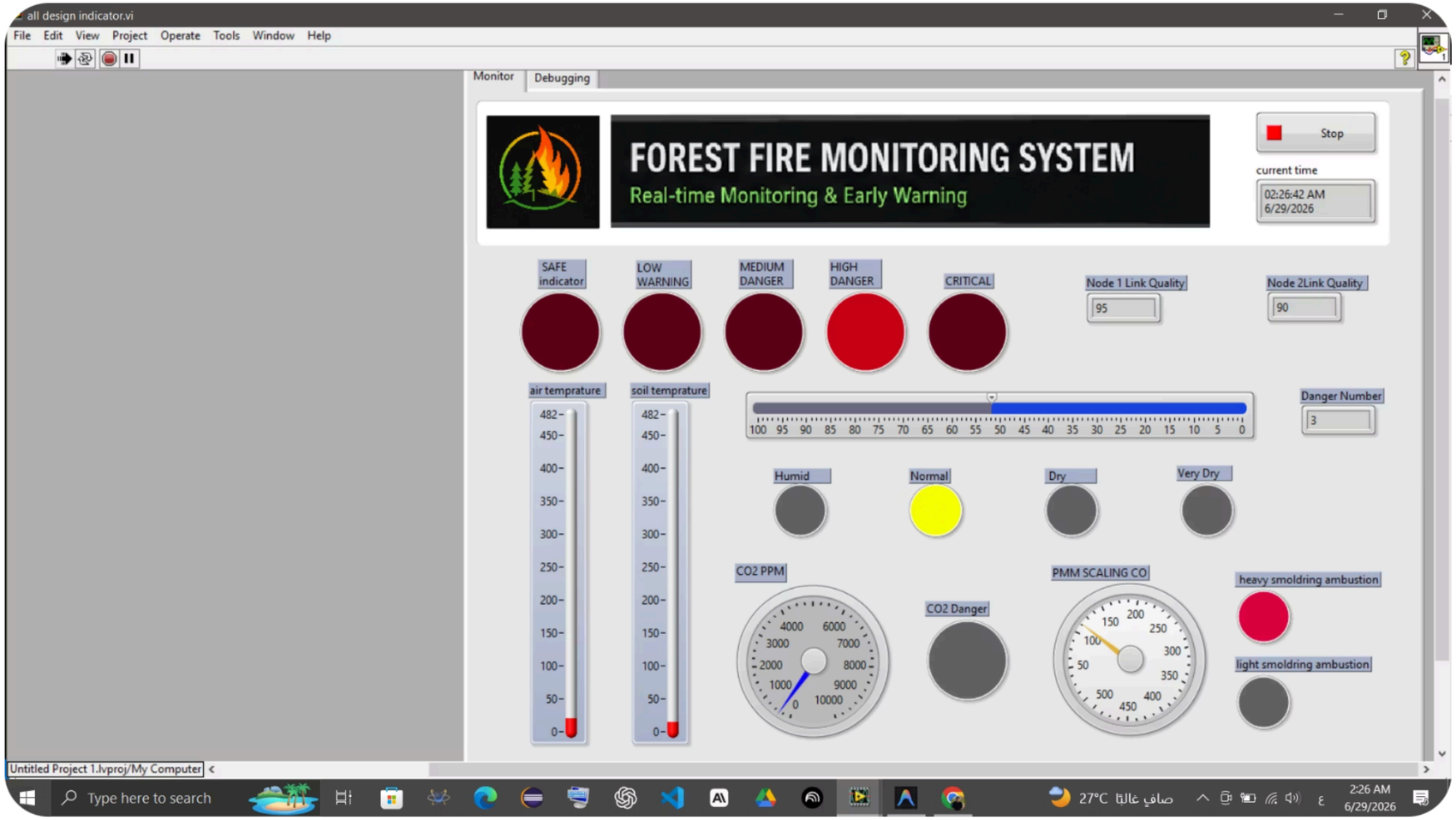


Front Panel Real-Time Monitoring Interface

Adaptive Sampling Rate

Danger Level	Status	Soil Moisture	CO_2 (ms)	CO (ms)	Temperature (ms)
0	SAFE	6 hour	15 minute	15 minute	15 minute
1	LOW WARNING	3 hour	5 minute	5 minute	5 minute
2	MEDIUM DANGER	1 hour	1 minute	1 minute	1 minute
3	HIGH DANGER	15 minute	15 second	15 second	15 second
4	CRITICAL	5 minute	2 second	2 second	2 second

Real-Time Monitoring Dashboard



Dashboard Features

Real-time Sensor Monitoring

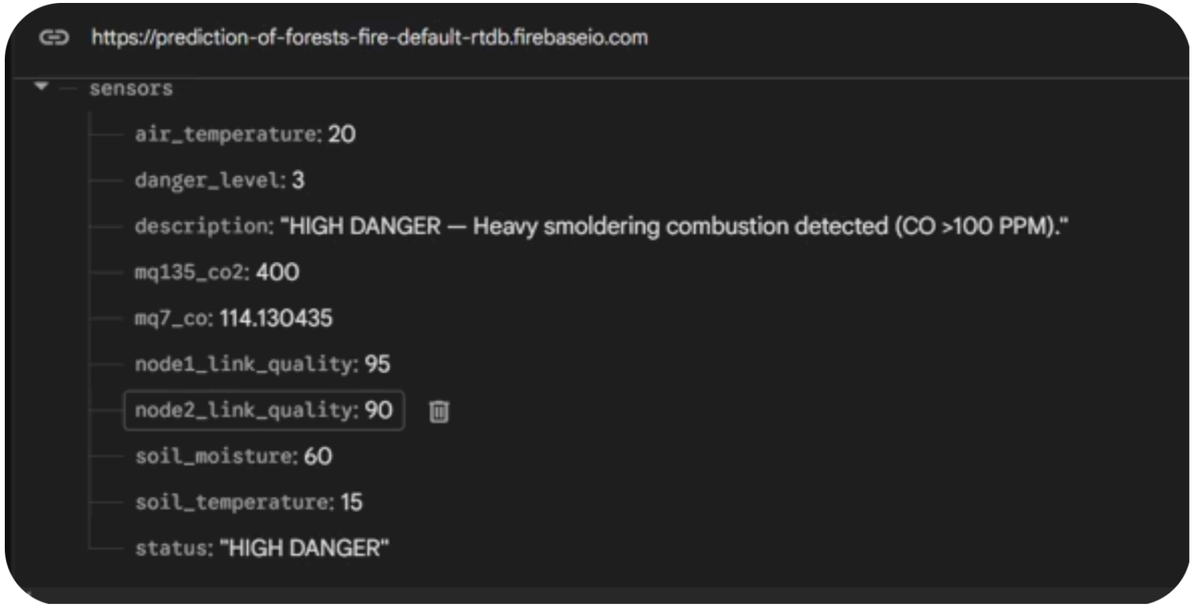
Live Risk Level Display

System Status Indicators

Automatic Data Refresh

User-Friendly Interface

"The LabVIEW dashboard provides continuous real-time monitoring of environmental conditions and fire risk status."



Firebase Realtime Database

Firebase stores time-stamped sensor readings and risk states. The cloud database enables remote access, persistence, and live synchronization across the system.

Real-Time Data Synchronization

Cloud-Based Storage

Remote Data Access

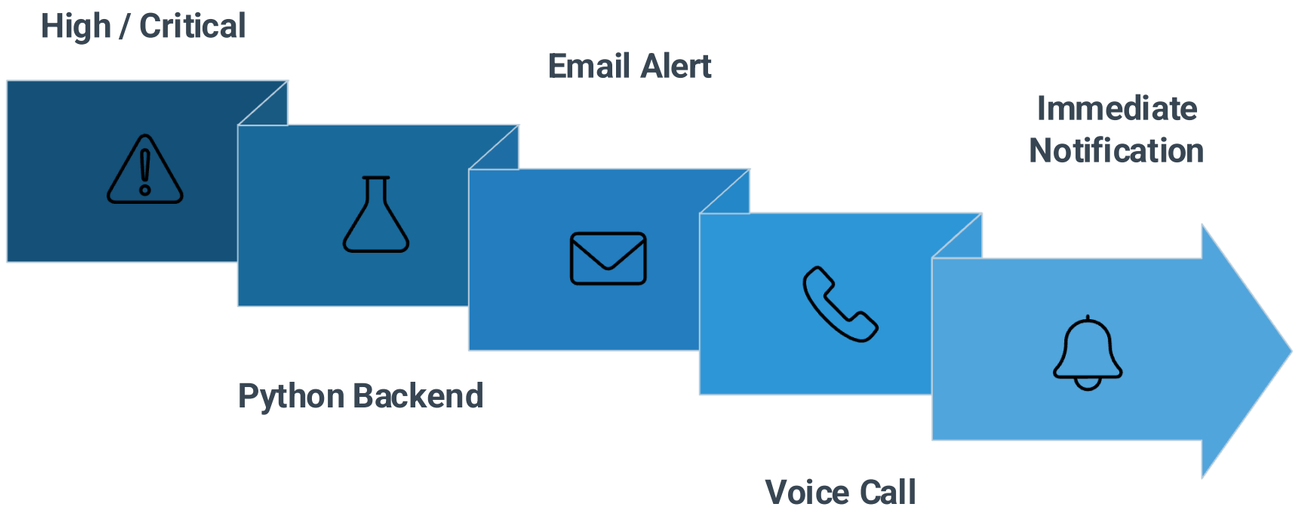
Automatic Updates

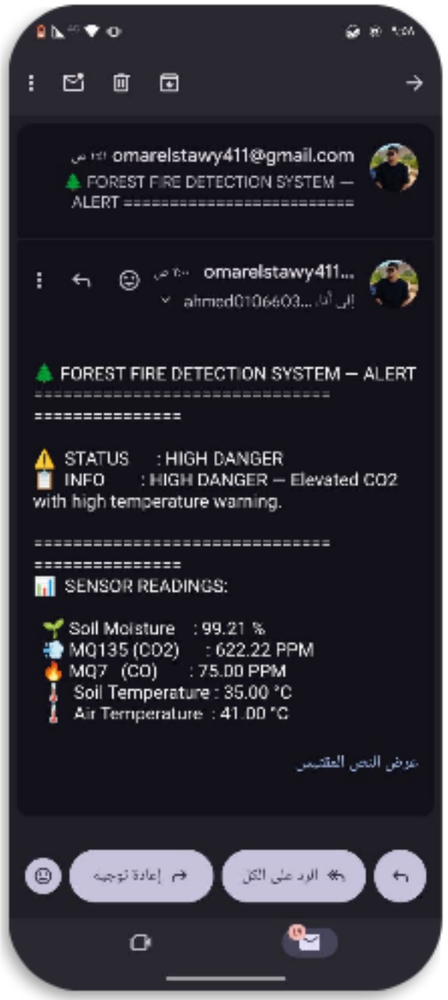
"Firebase provides real-time cloud synchronization, enabling remote monitoring and secure storage of environmental data."

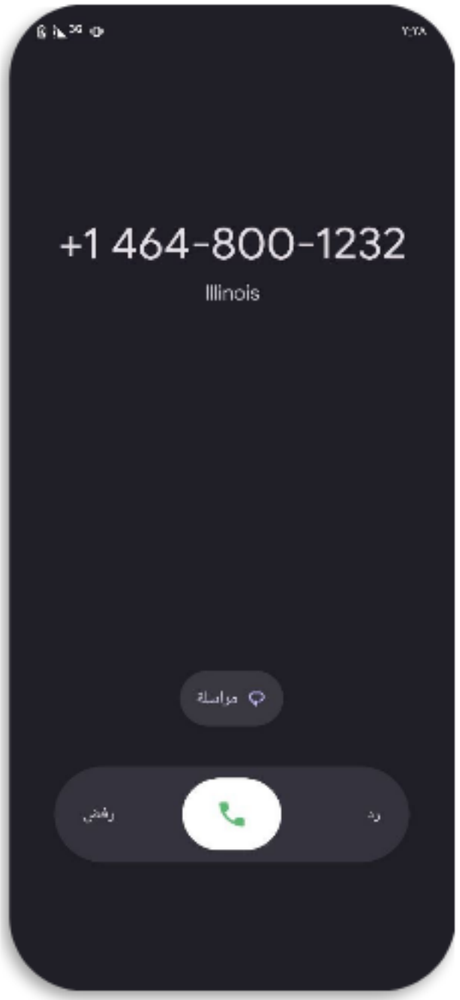
Alert & Notification System

Alert Features

"Automatic notifications ensure that critical fire events are reported immediately, enabling rapid emergency response."




omarelstawy411@gmail.com
FOREST FIRE DETECTION SYSTEM — ALERT
FOREST FIRE DETECTION SYSTEM — ALERT
STATUS : HIGH DANGER
INFO : HIGH DANGER — Elevated CO2 with high temperature warning.
SENSOR READINGS:
Soil Moisture : 99.21 %
MQ135 (CO2) : 622.22 PPM
MQ7 (CO) : 75.00 PPM
Soil Temperature : 35.00 °C
Air Temperature : 41.00 °C



The system detects high/critical risk, sends the event to the Python backend for analysis and routing, then triggers parallel notification paths (email and automated voice call) that converge to alert users immediately.

Automatic Email Notification

Automated Voice Call

Fast Emergency Response

Triggered for High-Risk Events

Real-Time Risk Warning

System Validation & Discussion

Validation Summary

- Successful sensor data acquisition
- Reliable WSN communication
- Stable TCP/IP communication
- Accurate fire risk classification
- Real-time LabVIEW visualization
- Successful Firebase synchronization
- Automatic email and voice call notifications

Accuracy

Correct fire risk classification across all test scenarios.

Real-Time Performance

Continuous monitoring with immediate dashboard updates.

Reliability

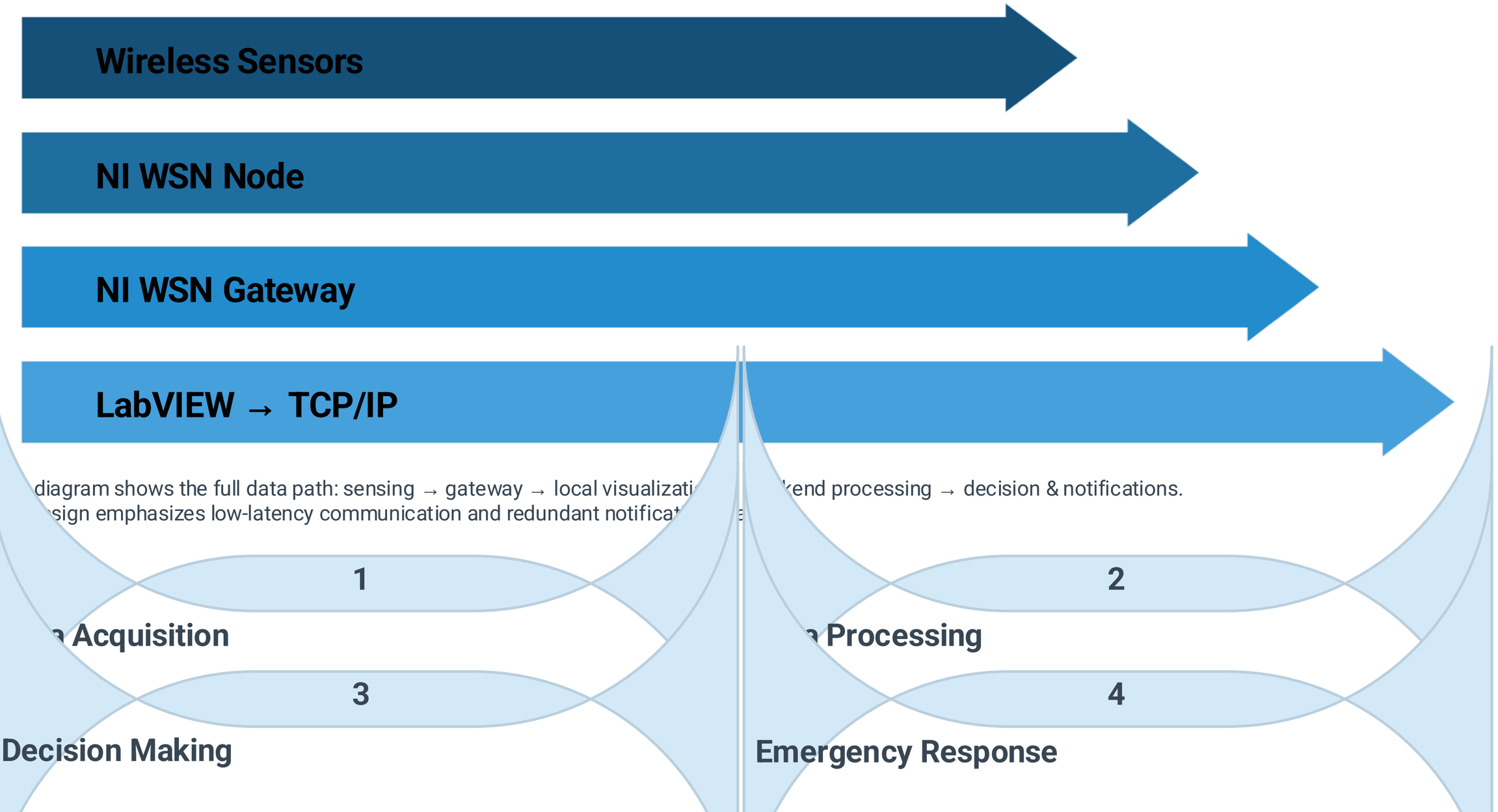
Stable communication between LabVIEW, Python, and Firebase.

Automation

Automatic cloud synchronization and emergency notifications.

"The experimental results demonstrate that the proposed system provides reliable real-time forest fire monitoring, accurate risk assessment, cloud integration, and automated emergency notification."

Complete System Workflow



Project Contributions & Conclusion

Wireless Sensor Network

Real-time environmental monitoring

LabVIEW Integration

Data acquisition and visualization

Python Backend

Fire risk analysis and classification

Firebase Cloud

Real-time cloud synchronization

Alert System

Automatic email and voice call notifications

Early Fire Detection

Supports rapid emergency response

Key Takeaways

- Multi-sensor fire monitoring
- Real-time wireless data acquisition
- Intelligent fire risk classification
- Cloud-based data storage
- Automatic emergency notification
- Reliable end-to-end system integration

"The proposed system demonstrates a reliable and scalable solution for real-time forest fire monitoring by integrating WSN, LabVIEW, Python, and cloud technologies."

Future Work

- Integrate Artificial Intelligence (AI) and Machine Learning (ML) models to predict forest fires before critical conditions occur and improve classification accuracy.

- Develop predictive analytics dashboards to analyze long-term environmental trends and support proactive decision-making.

- Integrate GPS modules and satellite remote sensing to accurately identify the location of fire incidents.

- Develop Android and iOS mobile applications for real-time monitoring, push notifications, and remote system management.

- Adopt LoRa/LoRaWAN communication to extend network coverage while maintaining ultra-low power consumption.

- Deploy UAVs (drones) equipped with thermal cameras to verify fire alarms and provide live aerial monitoring.

- Integrate solar energy harvesting techniques to increase sensor node lifetime and reduce maintenance costs.

- Enhance cybersecurity by implementing secure communication protocols, encryption, authentication, and intrusion detection mechanisms.

Thank You!